

Modifikasi *Route Discovery* pada *Ad-hoc on Demand Distance Vector (AODV)* Berdasarkan Level Konektivitas *One-Hop Node* di VANETs

Aviananda Dwirahma J, Radityo Anggoro, dan Supeno Djanali

Departemen Informatika, Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember (ITS)

e-mail: onngo@if.its.ac.id

Abstrak—*Vehicular Ad hoc Networks (VANETs)* merupakan pengembangan dari *Mobile Ad hoc Network (MANET)* dimana *node* memiliki karakteristik dengan mobilitas yang sangat tinggi dan terbatas pada pola pergerakannya. Ada banyak *routing protocol* yang dapat diimplementasikan pada VANETs, salah satunya adalah *Ad hoc On demand Distance Vector (AODV)*.

AODV merupakan salah satu *routing protocol* yang termasuk dalam klasifikasi *reactive routing protocol*, sebuah protokol yang hanya akan membuat rute ketika *node* sumber membutuhkannya. AODV memiliki dua fase, yaitu *route discovery* dan *route maintenance*. *Route discovery* digunakan untuk meminta dan meneruskan informasi rute yang terdiri dari proses pengiriman *Route Request (RREQ)* dan *Route Reply (RREP)*. Modifikasi pada Studi ini akan dilakukan pada proses *route discovery* berdasarkan level konektivitas *one-hop node*, yaitu dengan cara mengeliminasi jumlah *forwarding node* yang bertugas untuk mengirim ulang (*re-broadcast*) RREQ. Hal ini dilakukan agar dapat meningkatkan kinerja protokol AODV untuk mencari rute yang stabil dengan cara memodifikasi beberapa bagian dari mekanisme pengiriman paket RREQ. Dari hasil uji coba, AODV yang dimodifikasi berhasil meningkatkan nilai *Packet Delivery Ratio (PDR)* hingga 9,19%, penurunan nilai *Routing Overhead (RO)* hingga 2,17%, dan penurunan nilai *Forwarded Route Request (RREQ F)* hingga 31,62%.

Kata Kunci—AODV, *Forwarding Node*, NS-2, *One-Hop Node*, *Threshold*, VANETs.

I. PENDAHULUAN

SAAT ini perkembangan teknologi informasi dan komunikasi menjadi salah satu indikator kemajuan peradaban manusia. Salah satu teknologi yang membantu manusia dalam berkomunikasi dengan mudah adalah *Vehicular Ad hoc Networks (VANETs)*. VANETs merupakan suatu mekanisme yang dapat menghubungkan kendaraan satu dengan yang lainnya menggunakan jaringan nirkabel. VANETs dapat berguna pada banyak hal, seperti mengemudi secara otomatis, navigasi, pencegahan kecelakaan yang dapat meningkatkan keamanan berkendara, serta dapat mengurangi kemacetan lalu lintas [1].

Pada Studi ini diusulkan suatu mekanisme *routing discovery* pada *reactive routing* AODV untuk memperoleh rute berdasarkan level konektivitas *one-hop* pada VANETs. Penulis mengadaptasi cara kerja OLSR dimana hanya beberapa *node* yang terpilih sebagai *node* perantara untuk

meneruskan paket dan mengurangi *control packet* yang terlalu banyak [2]. *Node* tersebut hanya memilih beberapa *node* tetangga yang akan menjadi *node* perantara, dan hanya *node* perantara yang dapat meneruskan *Route Request (RREQ)*, sedangkan *node* lainnya tidak akan meneruskan (*drop*) paket tersebut. Metode yang digunakan untuk mendeteksi *node* tetangga menggunakan *HELLO messages* yang terdapat pada AODV yang dikirim secara periodik untuk mengetahui simpul tetangga pada suatu waktu [3]. Hasil akhir yang diharapkan adalah mengetahui perbandingan kinerja antara AODV dan AODV yang telah dimodifikasi diukur berdasarkan performansi *Packet Delivery Ratio (PDR)*, *Routing Overhead*, *End-to-End Delay*, dan *Forwarded Route Request (RREQ F)*.

II. ANALISIS DAN PERANCANGAN

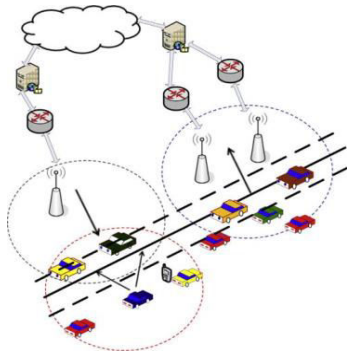
A. *Vehicular Ad-hoc Networks (VANETs)*

Vehicular Ad-hoc Networks (VANETs) merupakan pengembangan dari *Mobile Ad-hoc Network (MANET)* dimana pengembangannya difokuskan pada kendaraan (*vehicle*) yang dapat saling berkomunikasi maupun mengirimkan data. VANETs adalah sebuah teknologi baru yang memadukan kemampuan komunikasi nirkabel kendaraan menjadi sebuah jaringan yang bebas infrastruktur serta memiliki karakteristik mobilitas yang sangat tinggi dan terbatas pada pola pergerakannya. *Node* dalam jaringan dianggap sebagai *router* yang bebas bergerak dan bebas menentukan baik menjadi *client* maupun menjadi *router*. Protokol *routing* pada VANETs memiliki dua model yaitu protokol *reactive routing* yang membentuk tabel *routing* hanya saat dibutuhkan dan protokol *proactive routing* yang melakukan pemeliharaan tabel *routing* secara berkala pada waktu tertentu. Pergerakan *node* pada VANETs bisa berubah setiap saat dan terbatas pada rute lalu lintas yang dapat ditentukan dari koordinat peta. Hal ini membuat setiap *node* akan terus memperbarui informasi dalam tabelnya sesuai informasi dari *node* lain. Perubahan pergerakan pada VANETs menjadi salah satu permasalahan dalam pengiriman paket data sehingga dibutuhkan informasi jarak antar *node*, kecepatan dan *delay* transmisi [4]. Ilustrasi VANETs dapat dilihat pada Gambar 1.

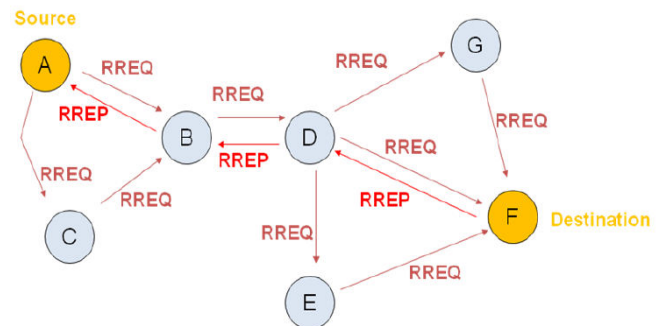
Tabel 1.
Parameter Simulasi

No.	Parameter	Spesifikasi
-----	-----------	-------------

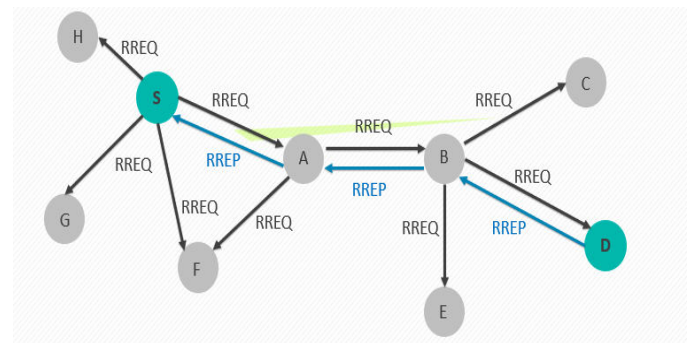
1	Network simulator	NS-2.35
2	Routing protocol	AODV
3	Waktu simulasi	200 detik
4	Area simulasi	1500 m x 1500 m
5	Jumlah Node	60, 150, 300
6	Radius transmisi	400m
7	Kecepatan maksimum	20 m/s
8	Protokol MAC	IEEE 802.11p
9	Model Propagasi	Two-ray ground



Gambar 1. Ilustrasi VANETs [5].



Gambar 2. Ilustrasi Pencarian Rute pada AODV [7].



Gambar 3. Ilustrasi Pencarian Rute pada AODV Modifikasi.

A. Ad-hoc On Demand Distance Vector (AODV)

Ad-hoc On Demand Distance Vector (AODV) adalah salah satu *routing protocol* yang termasuk dalam klasifikasi *reactive routing protocol*. Sebuah protokol yang hanya membuat sebuah rute saat dibutuhkan. AODV dikembangkan oleh C. E. Perkins, E.M. Belding-Royer dan S. Das pada RFC 3561.

Ada dua tahapan dalam AODV yaitu *route discovery* dan *route maintenance*. *Route discovery* memiliki dua pesan yaitu berupa *Route Request (RREQ)* dan *Route Reply (RREP)*. Sedangkan *Route maintenance* berupa *Route Error (RERR)*.

AODV adalah sebuah metode *routing* pesan antar *node* yang memungkinkan *node-node* tersebut untuk melewati pesan melalui lingkungannya ke *node* yang tidak dapat dihubungi secara langsung. AODV melakukan ini dengan cara menemukan rute yang bisa dilalui oleh pesan. Selain itu AODV juga memastikan rute ini tidak mengandung perulangan (*loop*), menangani perubahan rute, dan membuat rute baru apabila terjadi *error* [6]. Ilustrasi pencarian rute oleh AODV dapat dilihat pada Gambar 2.

B. Network Simulator-2 (NS-2)

Network Simulator (NS) adalah suatu *interpreter* yang berorientasi objek, dan *discrete event-driven* yang dikembangkan oleh University of California Berkeley dan USC ISI sebagai bagian dari proyek *Virtual Internet Testbed (VINT)*. NS yang banyak dikenal dengan NS-2 (versi 2) menjadi salah satu *tool* yang sangat berguna untuk menunjukkan simulasi jaringan melibatkan *Local Area Network (LAN)*, *Wide Area Network (WAN)*, tapi fungsi dari *tool* ini telah berkembang selama beberapa tahun belakangan untuk memasukkan jaringan nirkabel (*wireless*) dan juga jaringan *ad hoc* [8].

Pada Studi ini, NS-2 digunakan untuk melakukan simulasi lingkungan VANETs menggunakan protokol AODV yang

sudah dimodifikasi. *Trace file* yang dihasilkan oleh NS-2 juga digunakan untuk mengukur performa *routing* protokol AODV yang sudah dimodifikasi..

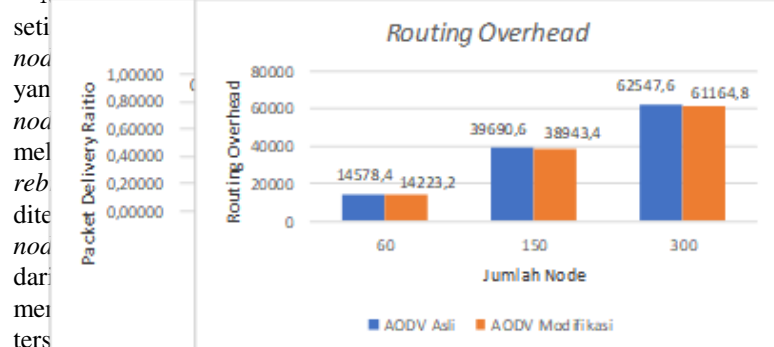
C. Simulation of Urban Mobility (SUMO)

Simulation of Urban Mobility (SUMO) merupakan paket simulasi lalu lintas yang bersifat *open-source* dimana pengembangannya dimulai pada tahun 2001. Dan semenjak itu SUMO telah berubah menjadi sebuah simulasi lalu lintas dengan kelengkapan fitur dan pemodelannya termasuk kemampuan jalannya jaringan untuk membaca *format* yang berbeda.

SUMO juga memungkinkan untuk mendefinisikan kendaraan dengan sifat tertentu seperti panjang kendaraan, kecepatan maksimum, percepatan dan perlambatannya. SUMO juga menyediakan pilihan bagi pengguna menentukan rute acak untuk kendaraan. Ada juga pilihan yang tersedia untuk model sistem transportasi umum, dimana setiap kendaraan datang dan berangkat sesuai dengan jadwal [9].

III. MODIFIKASI

Modifikasi akan diawali dengan pencarian jumlah tetangga



Gambar 4. Grafik Routing Overhead.

sebaliknya, maka *node* tersebut tidak akan meneruskan paket RREQ atau paket akan di-drop. Ilustrasi pencarian rute oleh AODV yang dimodifikasi dapat dilihat pada Gambar 3. Berikut algoritma modifikasi pada *routing protocol* AODV.

1. Specify threshold value
2. Calculate neighbor node
3. if (neighbour node < threshold value)
4. then drop RREQ packet,
5. else forward RREQ packet

IV. SIMULASI

Pada bagian ini dijelaskan mengenai lingkungan simulasi dan parameter simulasi untuk melakukan perbandingan performa AODV asli dan AODV modifikasi.

A. Lingkungan Simulasi

Terdapat 3 jenis lingkungan yang akan digunakan pada simulasi, yaitu 60 *node* untuk lingkungan jarang, 150 *node* untuk lingkungan sedang, dan 300 *node* untuk lingkungan padat.

B. Parameter Simulasi

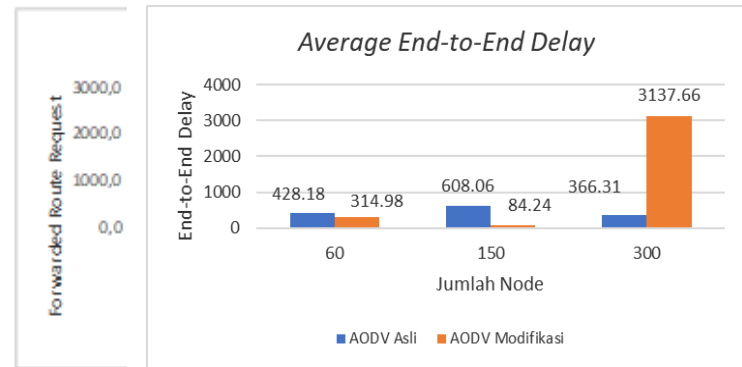
Untuk melakukan simulasi, digunakan *network simulator* NS-2.35 dengan protokol AODV. Skenario yang digunakan berukuran 1500m x 1500m. Keterangan lebih lengkap dapat dilihat pada Tabel 1.

V. HASIL UJI COBA

Pengujian dilakukan dengan menjalankan skenario yang disimulasikan pada NS-2. Dari simulasi tersebut dihasilkan sebuah *trace file* dengan ekstensi .tr yang akan dianalisis dengan bantuan skrip awk untuk mendapatkan PDR, E2E, RO, dan RREQ. Dilakukan pra uji coba untuk menentukan nilai *threshold* yang akan dijadikan sebagai perbandingan dengan *node* tetangga. Penentuan nilai *threshold* akan dilakukan dengan cara melakukan simulasi dengan *threshold* yang diambil dari angka satu hingga angka yang membuat nilai *Packet Delivery Ratio* (PDR), *Routing Overhead* (RO), dan *Forwarded Route Request* (RREQ F) pada AODV modifikasi turun secara drastis dibandingkan dengan AODV asli. Pra uji coba dilakukan selama 10 kali dan dilihat hasil metrik analisis yang terbaik dan muncul paling banyak pada 10 skenario tersebut. Hasil yang unggul tersebut akan dijadikan nilai *threshold* untuk diimplementasikan pada lingkungan uji coba. Dari hasil pra uji coba, didapatkan nilai *threshold* 9 untuk lingkungan 60 *node* (jarang), nilai *threshold* 21 untuk lingkungan 150 *node* (sedang), dan nilai *threshold* 45 untuk lingkungan 300 *node* (padat). Untuk perbandingan hasil pengujian dapat dilihat masing – masing pada Gambar 4, Gambar 5, Gambar 6, dan Gambar 7.

Berdasarkan grafik pada Gambar 4, dapat dilihat bahwa *routing protocol* AODV asli dan AODV yang telah dimodifikasi mengalami penurunan yang signifikan pada PDR. Pada lingkungan yang jarang dengan *node* berjumlah 60, menghasilkan perbedaan selisih PDR sebesar 0,04514, dimana terjadi kenaikan PDR sebesar 5,43% dan *routing protocol* AODV yang dimodifikasi unggul dalam hal PDR tersebut.

Pada lingkungan yang sedang dengan jumlah 150 *node*, menghasilkan perbedaan selisih PDR sebesar 0,02484, dimana terjadi kenaikan PDR sebesar 3,02% dan *routing protocol* AODV yang dimodifikasi unggul dalam hal PDR tersebut dari AODV asli. Pada lingkungan yang padat dengan jumlah 300 *node*, menghasilkan perbedaan selisih PDR sebesar 0,12988, dimana terjadi kenaikan sebesar 19,12% dan *routing protocol* AODV yang telah dimodifikasi juga unggul dalam hal PDR tersebut. Nilai rata-rata kenaikan PDR pada skenario *real* adalah sebesar 9,19%.



Gambar 6. Grafik End-to-End Delay.

Berdasarkan grafik pada Gambar 5, dapat dilihat bahwa *routing protocol* AODV yang telah dimodifikasi dan juga *routing protocol* AODV asli mengalami kenaikan *routing overhead*. Pada lingkungan yang jarang dengan jumlah 60 *node*, menghasilkan perbedaan selisih RO sebesar 2144, dimana terjadi penurunan RO sebesar 2,43% dan *routing protocol* AODV yang telah dimodifikasi unggul dalam hal RO tersebut karena menghasilkan RO yang lebih rendah dari AODV asli. Pada lingkungan yang sedang dengan jumlah 150 *node*, menghasilkan perbedaan selisih *routing overhead* sebesar 747.2, dimana terjadi penurunan RO sebesar 1,88% dan *routing protocol* AODV yang telah dimodifikasi unggul dalam hal RO tersebut dari AODV asli. Pada lingkungan yang padat dengan jumlah 300 *node*, menghasilkan perbedaan selisih RO sebesar 1382.8, dimana terjadi penurunan RO sebesar 2,21% dan *routing protocol* AODV yang telah dimodifikasi juga unggul dalam hal RO tersebut. Nilai rata – rata penurunan PDR pada simulasi *real* adalah sebesar 2,17%. Dapat dilihat pula bahwa AODV yang telah dimodifikasi menghasilkan RO yang lebih bagus atau dalam hal ini lebih rendah daripada AODV asli.

Berdasarkan grafik pada Gambar 6, dapat dilihat bahwa *routing protocol* AODV yang telah dimodifikasi dan juga *routing protocol* asli mengalami perubahan *forwarded route request* (RREQ F) yang signifikan. Pada lingkungan yang jarang dengan jumlah 60 *node*, menghasilkan perbedaan selisih *forwarded route request* sebesar 304.4, dimana terjadi penurunan RREQ F sebesar 14,20% dan *routing protocol* AODV yang telah dimodifikasi unggul dalam hal RREQ F tersebut karena menghasilkan RREQ F yang lebih rendah dari *routing protocol* AODV asli. Pada lingkungan yang sedang dengan jumlah 150 *node*, menghasilkan perbedaan selisih *forwarded route request* sebesar 683, dimana terjadi penurunan RREQ F sebesar 31,23% dan *routing protocol*

AODV yang telah dimodifikasi unggul dalam RREQ F tersebut dari RREQ F AODV asli. Pada lingkungan yang padat dengan jumlah 300 *node*, menghasilkan perbedaan selisih RREQ F sebesar 1370.6, dimana terjadi penurunan RREQ F sebesar 49,44% dan *routing protocol* AODV yang telah dimodifikasi juga unggul dalam hal RREQ F tersebut. Nilai rata-rata penurunan RREQ F pada skenario *real* adalah sebesar 31,62%. Dapat dilihat pula bahwa AODV yang telah dimodifikasi menghasilkan RREQ F yang lebih bagus atau dalam hal ini lebih rendah daripada RREQ F asli dengan jumlah selisih RREQ F yang cukup signifikan.

Berdasarkan grafik pada Gambar 7, dapat dilihat bahwa rata-rata E2E antara *routing protocol* AODV asli dan AODV yang telah dimodifikasi mengalami perubahan yang fluktuatif. Pada lingkungan yang jarang dengan jumlah 60 *node*, terjadi perbedaan selisih *end-to-end delay* sebesar 113.2008 ms antara *routing protocol* AODV asli dengan *routing protocol* AODV yang telah dimodifikasi, dimana *routing protocol* AODV yang telah dimodifikasi unggul dalam hal E2E tersebut. Sedangkan pada lingkungan sedang dengan jumlah 150 *node*, terjadi perbedaan selisih *end-to-end delay* sebesar 523.823 ms antara *routing protocol* AODV asli dengan *routing protocol* AODV yang telah dimodifikasi, dimana *routing protocol* AODV yang telah dimodifikasi lebih unggul dalam hal E2E. Pada lingkungan yang padat dengan jumlah 300 *node*, terjadi perbedaan selisih E2E sebesar 2771.35632 ms antara *routing protocol* AODV asli dengan *routing protocol* AODV yang telah dimodifikasi, dimana *routing protocol* AODV yang asli jauh lebih unggul dalam hal *end-to-end delay* tersebut.

Jika ketiga lingkungan tersebut dibandingkan, memang pada lingkungan yang jarang, sedang, maupun padat tidak selalu lebih unggul dalam hal E2E. Hasil rata – rata E2E tidak dapat dianalisis karena terjadi fluktuasi dan tidak stabil. Hal ini dikarenakan waktu *delay* tergantung dari rata – rata waktu paket yang terkirim. Semakin banyak paket yang terkirim, maka semakin beragam *delay*nya.

VI. KESIMPULAN DAN SARAN

Setelah dilakukan uji coba, didapatkan kesimpulan sebagai berikut:

Penulis wajib menghindari artikelnya dari kemungkinan plagiarisme.

- 1) AODV yang dimodifikasi sudah berhasil membatasi jumlah *forwarding node* yang bertugas untuk *rebroadcast* paket RREQ.
- 2) Dampak pembatasan *forwarding node* terhadap performa protokol AODV secara keseluruhan adalah rata – rata kenaikan *Packet Delivery Ratio* (PDR) sebesar 9,19%,

rata – rata penurunan *Routing Overhead* (RO) sebesar 2,17%, dan rata – rata penurunan *Forwarded Route Request* (RREQ F) sebesar 31,62%.

Adapun saran-saran yang diberikan untuk pengembangan Studi ini adalah sebagai berikut:

- 1) Lebih banyak uji coba yang dilakukan untuk mendapatkan hasil yang lebih akurat.
- 2) Perhitungan jumlah *node* tetangga yang lebih dinamis pada *routing protocol* AODV. Contohnya saat *node* tetangga menjauh dari jangkauan, maka ada mekanisme untuk memperbarui *list* simpanan *node* tetangga sehingga perhitungan jumlah *node* tetangga lebih akurat.
- 3) Menambahkan aspek lain untuk melakukan pembatasan *forwarding node* yang meneruskan paket seperti arah, kecepatan, dan energi.

DAFTAR PUSTAKA

- [1] “VANET - Vehicle Ad hoc Network,” [Online]. Available: http://comp.ist.utl.pt/~rnr/WSN/CaseStudies2007-no/WSN_Transportation/. [Diakses 15 November 2017].
- [2] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum dan L. Viennot, “Optimized Link State Routing Protocol for Ad hoc Networks,” IEEE, Lahore, 2001.
- [3] S. N. Ferdous dan M. S. Hossain, “Randomized Energy-Based AODV Protocol for Wireless Ad-Hoc Network,” IEEE, Dhaka, 2016.
- [4] R. Brendha dan V. S. J. Prakash, “A Survey on Routing Protocols for Vehicular Ad hoc Networks,” IEEE, Coimbatore, 2017.
- [5] R. G. Engoulou, M. Bellaiche, S. Pierre dan A. Quintero, “VANET Security Surveys,” *Computer Communication*, vol. 44, p. 2, 2014.
- [6] R. Brendha dan V. S. J. Prakash, “A Survey on Routing Protocols for Vehicular Ad hoc Networks,” IEEE, Coimbatore, 2017.
- [7] M. Iqbal, M. Shafiq, H. Attaullah, J.-G. Choi, K. Akram dan X. Wang, “Design and Analysis of a Novel Hybrid Wireless Mesh Network Routing Protocol,” p. 22, January 2014.
- [8] P. Meeneghan dan D. Delaney, “An Introduction to NS Nam and OTcl scripting,” April 2004.
- [9] D. Krajzewics, J. Erdmann, M. Behrisch dan L. Bieker, “Recent Development and Application of SUMO,” *International Journal On Advances in Systems and Measurements*, p. 128, December 2012.